

chat2 patch testing

These are the detailed results from Michigan's stress-testing of the chat2 tool. The last six weeks have seen some exciting developments from Charles Hedrick and Dave Haines in regards to this tool (see sak-13274). We have seen some production performance issues specifically related to the chat tool. The recent performance upgrades go a long way towards addressing these issues. The current sakai 2.4 chat2 tool appears to execute SQL in the database long after user's sessions have ended. The current build is also very CPU intensive on the database throughout a chat session. The recent patch alleviates the first problem entirely, and diminishes database CPU activity a great deal.

Our test scenario uses a number of sakai tools at regular usage levels. Rwiki, forums, gradebook, resources, dropbox, messages, syllabus, assignments, discussion, schedule, and announcements are all exercised to some extent. We also have two scripts that exercise chat. The first logs on to one of several thousand sites, sends 4 chat messages over the course of 90 seconds, and logs off. The second performs the same steps, but emulates 50 users doing this in a single site. This situation of many users logged into a single site, mostly listening but occasionally sending messages, has been the cause of performance issues here at Michigan. It is this second script that highlights the difference in chat performance. Our test runs for 70 minutes.

Bear in mind that 95 seconds of "wait" time is hard-coded in our chat script, so total script response time will never be faster than 95 seconds. Also note that in addition to this wait time, total script response time includes time to log on, select a course, open the chat room, send 4 messages, and log out. [Here](#) is the response time graph for the current production build, [here](#) is the graph for the updated build. In both graphs, the upper-most plot indicates the script emulating 50 users in a single chat room. Notice the gradual degradation in response times for both builds. Essentially, the room is slower to load and refreshes take more time as the number of chat messages increase. However, the degradation is much more severe in the current production build.

One of the interesting data points from the previous build was the number of database SQL executions that occurred after the test was finished. [Here](#) is the graph of execution count for the current 2.4 production build, and [here](#) is the graph for the patched build. Notice the gradual taper off of database executions in the production build, whereas database activity drops off immediately after the test ends in the patched build. This is evidence that persistent beans are being destroyed as they should, per Charles' comments in the jira. Also, the patched build is less CPU intensive. [Here](#) is the DB CPU graph for the production build, and [here](#) it is for the patched build.

Any institutions experiencing performance problems with the current Sakai 2.4 chat2 tool or thinking of implementing it for the first time should consider the sak-13274 patch.

Please contact me with any questions or concerns.

- Chris

ckretler@umich.edu